

Embedded Energy Management System for the ICT Saving Energy Consumption

Francisco Maciá-Pérez, Diego Marcos-Jorquera, Virgilio Gilart-Iglesias,
Juan Antonio Gil Martínez-Abarca, Luis Felipe Herrera-Quintero,
and Antonio Ferrándiz-Colmeiro
*Computer Science Department. University of Alicante
Spain*

1. Introduction

The importance of Information and Communication Technologies (ICT) in all areas of human activity in today's world is an indisputable fact. In the last years, there has been an exponential increase of the use of these technologies within the society, from its professional use in enterprises and organizations to its personal use in playful and everyday activities at home. In addition, the new ICT paradigms evolution together with the growing use of Internet have caused the apparition of new business models that require complex systems in order to support them, available 24 hours per day 7 days per week, with better quality of service, etc.

However, this growing use of ICT technologies together with the requirements of emerging business models is converting these technologies in one of the main responsible of the worldwide energy consumption increase. In this way, (Gartner press, 2007) determines that the emission rate of CO₂ originated from the ICT consumption is the 2% and predict that this energy consumption will grow in an exponential way in the next years if solutions are not adopted.

In fact, one part of this consumption is due to an inefficient use of the ICT technologies. According to the study described in (Mines et al., 2008), a great number of the ICT managers know the necessary measures that they have to realize in order to obtain a energy saving produced by the use of ICT in their organizations, however, usually this measures are not applied if they do not mean an economic benefit for the business. One of the main reasons of the inadequate energy consumption of ICT listed in the study is the lack of awareness of the users in relation to this energetic problem that involve an incorrect use of the ICT infrastructures. Some examples of this uses are to leave power on Personal Computers (PC's), printers, servers or network devices when is not necessary.

There is the paradox that one of the solutions with more repercussion nowadays in order to optimize the energy consumption of the ICT is the use of the same ICT. This approach is one of the main proposals of the European Union (Commission European Report, 2008) that pretend to promote an efficient use of the energy consumption through the use of the Information and Communications Technologies.

In consonance with this approach, our proposal consists of providing embedded IT management services in physical network devices (generally, small sized devices with simple services and low energy consumption), so that, in order to deploy those services, it is enough to select the specific device providing the service, and connecting it to the communications network. The device itself will obtain the minimum information required to activate the initial set up and, once this has been completed, execute the management tasks with minimal human intervention.

Obviously, from a functional point of view the services offered by these devices are totally compatible with the traditional network services and therefore their integration and interoperability are ensured.

By way of illustration and with the aim of arguing the motivating of the proposal, we suggest a specific management service that we named Energy Management System (EMS): a service for the ICT systems monitoring and consumption control of these same systems doing that the ICT resources will be available only when they are necessities (in a proactive or scheduled way). Thus it will be possible to avoid processing and consumption during the downtimes. The goal of this service is to reduce and to optimize the energy consumption of the ICT infrastructures.

The basic function of the service will be to indicate to the embedded EMS device (eEMS) which equipment and which service or services of those equipments we wish to check in order to reduce the energy consumption. These actions will be done according to system global load or of the requirements defined by the user or system administrator.

In the following sections we provide a review of the current state of the art of the technologies involved; a description of the EMS service, hardware and software structure of the device in which it is embedded; the specification of the application protocol and its implementation as Web Service embedded in a specific network device and the test scenario in order to validate the proposal; and, finally, the conclusions on the research and the current lines of work.

2. Background

Increasing in the energy consumption has turned into a global problem. EU has ordered to the member states and industry to use the ICT to increase the energy efficiency as a mode to fight against climatic change and drive to economy recovery. According to European Union forecasts, through the ICT, the CO₂ emissions can be reduced up to 15% in 2020. For achieve this purpose, the saving energy is based on two mainly ways. On the one hand, to make aware population about how to use the energy. On the other hand, an improvement in control and management of the energy use in industries, offices and public places. In this document it is recommended that the ICT industry itself could be the pioneer reducing their own CO₂ emissions near to 20% in 2020 (European Union, 2008).

The majority of the proposals in order to reduce the ICT energetic consumption are focused on getting better design of the devices architectures. In (Moshnyaga & Tamaru, 1997) different design techniques of ICT devices architectures are described with the aim of reducing the energetic consumption of these devices. In this way the Green Grid (The green grid, 2009) is focused on the best practices and management approaches for lowering data centers energy consumption. The Department of Energy of USA released the Server Energy

Measurement Protocol (EnergyStar, 2009) that establishes a procedure for attaching an energy usage measurement to existing performance measurements for servers. Another approach (Lawton, 2007) very used nowadays to reduce the ICT energetic consumption is the virtualization. This proposal is originated from the hypothesis that the majority of the servers in the data centers are working to the 20% of its capability. The use of virtualization systems such as VMWare enables to execute virtual machines inside an only server, making good use of its processing capability.



Fig. 1. Google's Datacenter distribution.

In the same way, another alternative that takes an advantages of the virtualization for the reduction of power consumption of the DataCenters, as which it is produced by TIC's elements and its infrastructure(e.g refrigeration systems), basically it is the geographical distribution of DataCenters, under climatic zones that allows in dynamic way, move the computation to some places, where there exists a better conditions of temperature and also a places where the electricity's fees are lowest (Follow the moon). This approach is not oriented directly for the computation, besides, only this is applicable to a very big companies as Google.

However, in complex ICT environments with high availability requirements (replication, load balancing and clusterization), the proposals described previously are not enough to reduce the energetic consumption because the system management is not contemplated in a global way.

The use of embedded devices in order to provide services in a distributed environment is other of the solutions that allow decreasing the ICT infrastructures energetic consumption. In this sense, many of these devices include the Power over Ethernet (PoE) technology. This technology allows providing energy to the devices trough of Ethernet wire (Deuty, 2004).

On the other hand, there are many proposals in order to monitor and control the energy consumption trough of ICT tools and applications. In (Pietilainen, 2003) several of these tools are described. These tools use emergent technologies such as Internet and distributed systems to control and to supervise energy consumption. This kind of tools is oriented to inspect the general energetic consumption in the buildings, and although they could be used to control the ICT specific consumption, these tools and applications do not include

features of proactive management, autonomy and inattention to optimize the consumption. In these cases, the person that manage the application is who once analyzed the information obtained has to take the decision and to execute it himself in order to optimize the consumption. In addition, these tools have to be executed in PC's, servers or more complex systems, and therefore, add an increase of the energetic consumption.

The early researches about the energy management consumption were mainly focused on embedded and notebook systems. In these studies, the way of manage dynamically the energy for extending battery life is based on switching devices to lower-power modes when there is a reduced demand of services. Static strategies of energy management can lead to poor performance or unnecessary energy consumption when there are wide variations in the rate of requests of services (Ren et al., 2005). Some researches have augured that operating systems should be able to both implement energy-conservation policies and manage power for server applications at the system level (beini et al., 1998). In (Lien et al., 2004) a system for saving energy in a web server clusters has been proposed by using dinamic server management. So, architecture for Dynamic Web-Server has beens presented for resources management in a server cluster. The goal was to allocate different numbers of servers for different service rates in a way that automatically adapts the server cluster to the Web requests and improves the energy efficiency. According to these proposals, in (Lien et al., 2004) a system for estimation of the energy consumption of streaming media centers has been proposed. All of the mentioned studies show the importance of to achieve saving energy consumption, specially, when the number of machines wired in networks is very high.

The use of network management systems can help to automate the maintenance activities, allowing an efficient use of the network resources, and to be used to reduce the energy consumption. The first open standards which attempted to address problems of ICT management in a global manner were SNMP and CMIP (RFC project, 2009), proposed by the IETF (Internet Engineering Task Force); both protocols being principally oriented towards network monitoring and control. The main inconvenience of these administration models was their dependence on the platform.

The use of multi-agent systems for computer network management provides a series of characteristics which favour automation and self reliance in maintenance processes (Du et al., 2003) (Guo et al., 2005). The creation of projects such as AgentLink III, the first Coordinated Action on based on Agents financed by the 6th European Commission Framework Programme, is a clear indicator of the considerable degree of interest in research into software agents.

In areas where automated handling of information and those where several devices are involved, such as industrial processes or domotics, there has been a trend in the development of autonomous management towards architectures designed for services for embedded systems (Topp et al., 2002) (Jammes et al., 2005). This final framework includes monitoring systems developed by third parties but residing with the client, who is responsible for their control and management. Along these lines we find proposals such as NAGIOS (NAGIOS, 2009), MON (MON, 2009), MUNIN/MONIT (MUNIT, 2009) (MONIT, 2009) or nPULSE (nPULSE, 2009) generic monitoring systems for network services for linux, with Web interface, highly configurable and based on open code which monitors the availability of network services and applications. The disadvantage of these proposals is based on the complexity of their installation and configuration in environments without

qualified system administrators, in addition to the complex systems and infrastructures required for their implementation.

The approach described in this research work is presented as a solution that bring together the advantages of the current network management systems oriented to the control of ICT energetic consumption together with the use of embedded devices that minimize the consumption of these management systems.

3. Energy Management Service

The main goal of the EMS is to manage the power on or power off of a set of elements in a communications network in terms of a planning or in a proactive manner, analyzing the status of the system that is managed.

The eEMS is the version of the management service that has been implemented in Web Service, and it has been embedded in a network device (known as eEMS Device) designed for this purpose (see fig. 2). This device is small in size, with low consumption, robust, transparent to existing ICT infrastructures and with minimum maintenance required from the system administrators.

The system administrator informs the eEMS device, by means of its *interface agents*, which of the network components require a power management. The eEMS device has sufficient knowledge of each device to carry out this task. This knowledge is included in *management agents* displaced to the device for this purpose. The *management agents* implement specific protocols for power on devices, as the Wake on LAN (WoL) standard, or for power off, as the shutdown in SNMP. In some cases, to take the decision to power off or power on a device, they utilize a set of *monitoring agents* that analyze applications, services or network traffic. In this way, if the device receives a request for manage a set of devices, it will request the adequate *monitoring agents* and *management agents* in a self sufficient manner in order to carry out this work. The *management and monitoring agents* are enough flexible to adapt to the possible different scenarios.

Thus, the eEMS device represents the core of the system. Figure 2 shows a diagram of the main elements and actors involved in the service, together with the existing relation between them. We may synthesise these as: eEMS Device, Network Components, Discovery Service, EMS Center, EMS Clients, a set of Software Agents and the EMS application protocol (EMSP). These elements shall subsequently be described in greater detail.

The eEMS device, as has been seen, is the cornerstone of the energy management service. It is designed in order to act as a proxy between the Wide Area Network (WAN) and Local Area Network (LAN) to which it provides support. This device provides a container in which different agents and applications ensure that the service can be executed.

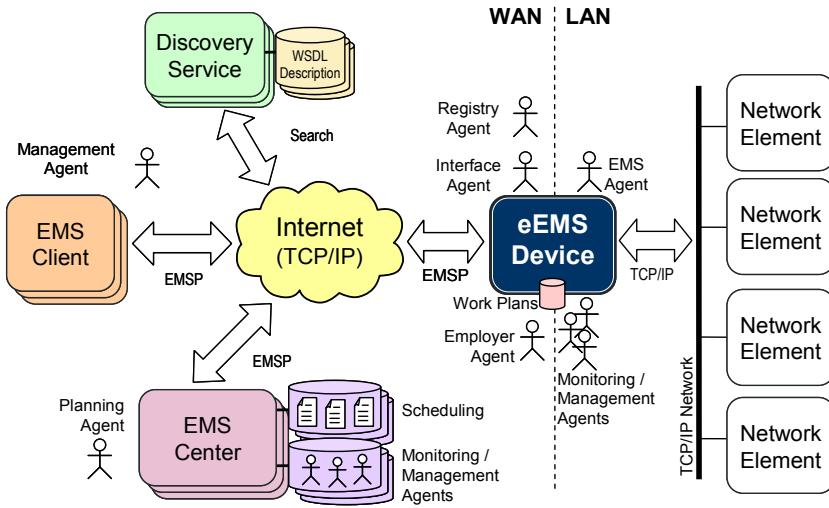


Fig. 2. Organization of functional elements of the EMS service.

In the proposal implementation, the device interface with the system administrators and with other management devices or management equipments is provided by agents acting as embedded Web Services (see *interface agent* in figure 2). From a functional point of view, this is the reason why an eEMS device can be taken into account, simply, as if it were a Web Service. In this way, an eEMS device is responsible for collecting the management request from the WAN. These requests are based on EMSP protocol and encapsulated in SOAP (Simple Object Access Protocol) messages when they are sent to the Web Service Interface.

The Network Components are the goal of the network monitoring service and comprise all those devices connected to the TCP/IP network. This include PC's, servers, printers, routers and, in general, any device susceptible to power on or power off in a remote manner.

The Discovery Service comprises a standard Universal Description, Discovery and Integration (UDDI) registration service. It is responsible for maintaining the pages describing the EMS services in Web Service Description Language (WSDL) format, as well as facilitating that information to the clients wishing to access the service.

EMS Centers usually act as automated control panels for the eEMS devices distributed through Internet. This control is implemented through the *planning agents* who carry out, execute and verify all the previously established tasks on the eEMS devices. EMS Centres are also responsible for managing the repository of *monitoring and management agents* with the know-how of each device management. Although in large installations it is recommended that management and scheduling services are included, the existence of an EMS centre is not essential. Likewise, although each EMS centre can manage around a thousand eEMS devices, it is possible to use the number of EMS centers considered appropriate, and it is possible to create one hierarchy with these elements.

EMS Clients, through the *EMS agents*, provide the user with access to the EMS Centre (in order to manage work plans or query log files) and to the eEMS Devices (in order to manage particular devices). These clients are not necessary for the normal operating system; however, they avoid physical movements of the system administration staff.

Software agents. System functionality has been defined as a distributed application based on software agents, because this approach intrinsically includes aspects such as: communications, synchronization, updates, etc. Among the agents that have been defined in the system, the most important are the agents placed in the eEMS device, and as a result, they comprise the system core. Of these last agents, the *interface agents* are of prime importance as they allow the device to provide its functionality to external elements (see section).

The EMS protocol (EMSP) is a request-response application level protocol using SOAP messages. This protocol is used by the different system components in order to communicate between each other. In fact, as the application has been designed as a set of software agents, the protocol will be used by the software agents to communicate with each other (see section 5).

4. Software Agents

The software agents do not constitute a conventional multi-agent system because a generic context has not been defined for them, they do not use standard agent communication languages and they do not work collaborating to achieve a general target which is used by the agents to take its decisions. In fact, the set of software agents implement part of the functionality of a distributed application which has been designed to provide a network service; in this case, the monitoring service. The reason why agent approach is used lies in its simplicity to design distributed applications and to take into account aspects such as communication, mobility or software updates.

Each eEMS device comprises a set of agents that implement its interface with the system administrators or with others system elements (EMS clients or EMS centers). In order to guarantee the system's compatibility with a large range of technologies, several interface agents have been implemented. In this way, the *interface agent* provides a matching interface with Web Services-based applications. The *interface agent* can identify commands based on *EMSP protocol* and, from these commands, schedule the eEMS device work plan. *EMS agents*, *management agents* and *monitoring agents* are another type of agent placed in the eEMS device and designed to perform the energy management service. The first type of agents ensures execution of the scheduling, delegating the specific monitoring task to a *monitoring agent* and the specific management task to a *management agent*. In addition to these core agents, other agents are included in each eEMS device in order to perform auxiliary tasks. Thus, the *register agents* undertake to check the monitoring service in a Discovery Service; and the *employer agents* are responsible for locating the *management agents* or *monitoring agents* required by the eEMS device to carry out its task. These agents are mobile agents that, initially, can reside in an agent farm located in an EMS Centre.

CMD	ACTION	ARG	FUNCTION
SET	MODE		Reports the current operation mode.
		PASSIVE [port]	Sets the passive mode and, optionally, the listening port number.
		ACTIVE <ip> [:port]	Sets the active mode, specifying the EMS center's IP address and port number.
	RUN		Reports the current EMS service state.
		<STARTS STOP>	Starts or stops the EMS service.
GET	SCHDL		Returns the list of scheduled tasks in the device.
	STATUS	[<host>[:port] [<service>]]	Returns the status of a specific service or a set of services.
PUT	SCHDL	<schdl-table>	Adds a task or a set of tasks to the scheduling.
MONITOR	ON	<host>:<port> <time>	Establishes a monitoring rule for the address <host>:<port>, establishing the polling time in seconds
		<service> [arguments]*	and the monitor that will be utilized as well as the arguments that this require.
	OFF	<host>:<port> <service>	Cancels a monitoring rule.
ALERT			Send an error alert.

Table 1. Main instructions of the EMS protocol.

Besides the agents located in each eEMS device, the distributed application is completed by other auxiliary agents located outside the device which, while not being crucial to the service, serve to make it more functional. As a result, the *client agents* reside in an EMS Client and are responsible for providing an appropriate interface for the administrators so that they can access the EMS Centre or an eEMS Device from any node connected to Internet. The *planning agents* reside in the EMS Centers and undertake the planning management of eEMS Devices.

5. EMS Protocol

The system agents, implemented in our prototype, communicate with each other by means of messages containing instructions capable of interpreting and executing. These instructions, together with their syntax and its pertinent response, come defined by the EMS Protocol or EMSP. When the agents specifically behave as Web Services, these commands will be incrustated inside the request and response SOAP messages. Web Services has been selected like communication protocol because it is an interoperable specification and that it permits to decouple totally the distinct actors of the system.

The EMS Protocol (EMSP) is a request-response application level protocol which gathers all monitoring service functionality through a set of instructions. The protocol has been defined as a request-response text-based application protocol. This enables it be easily adapted to different models, such as client-server (over basic protocols like HTTP, SMTP or telnet) and SOA (over protocols like SOAP).

The sequence diagram in figure 3 shows the basic service operation and the communication between the system software agents. The diagram comprises two blocks and is executed constantly in parallel mode. In the first block the device *interface agents* are on standby for

requests (from a *Planning Agent* or directly from a *Client Agent*). When the interface agents receive a monitoring request, they add the task to the Work Plan database of the eEMS device. The second diagram block corresponds to the execution of the programmed tasks. In this case the *EMS Agent* is constantly checking the Work Plan database and selecting the suitable *Monitoring Agent* and the *Management Agent* to carry out the requested tasks.

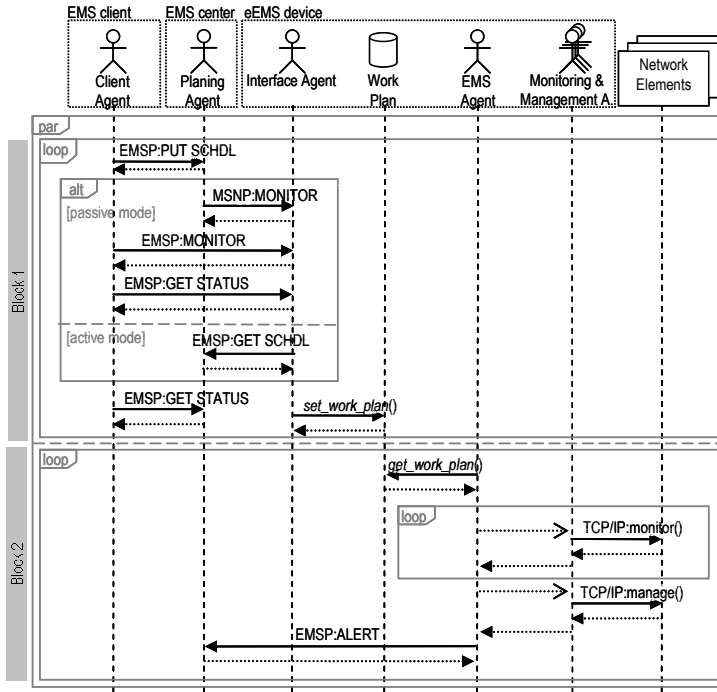


Fig. 3. Sequence diagram of the EMS main functionality.

Although it is not shown in this diagram, there is also a third block which concerns the contracting of the *Monitoring Agents*. When there is not a *Monitoring* or *Management Agent* able to deal with the service requested, the *EMS Agent* and the *Interface Agents* who have detected this lack may make a request to the *Employer Agent* programming it into its Work Plan. The *Employer Agent* then undertakes to obtain the *Monitoring Agents* required by the device. This agent is responsible for negotiating and validating the whole process. The *Monitoring & Management Agents* are mobile agents located in the agent repository in the EMS Centers.

6. eEMS Device Implementation and Test Scenario

In this section the implementation of an eEMS prototype device is presented (fig. 4). The hardware platform chosen for the prototype development is a *Lantronix Xport® AR™* device which has a 16 bit *DSTni-EX™* processor with 120MHz frequency reaching 30MIPS respectively (figure 4 shows an image of an eEMS device prototype connected to the service

network). The various memory modules provided by this device undertake specific tasks according to their intrinsic features: the execution programmes and the data handled by the device SRAM memory reside in the (1,25MB); the ROM memory (16KB) holds the system start up application and, finally, the flash memory, with 4MB, stores information which though non-volatile, is susceptible to change, such as the set up of the eEMS device or the system applications which may be updated. These capacities are sufficient for the memory requirements of the software developed for implementing the protocol.

Among other I/O interfaces, the device has a Fast Ethernet network interface which allows suitable external communications ratios. In addition, in order to ensure the correct system operation, there are several auxiliary elements such as: a watchdog which monitors the CPU and prevents it from blocking; and a PLL frequency divider required to set up the frequency of the system clock, with an adjustable clock signal (CLK) to optimise consumption or performance according to needs.

As a real time operating system, the device incorporates version 3 of the *Lantronix OS, Evolution OS™*. Through a confidentiality agreement with *Lantronix*, we have had access to the different modules of the system. Given the space restrictions, this has been crucial to develop a made-to-measure version of this OS. Salient elements of this version include, a TCP/IP stack together with several client-server application protocols (HTTP, TFTP, SNMP and telnet).

In the service layer, the implementation process has been conditioned by the limited characteristics of *XPort AR* device. Three service blocks are implemented: the middleware that provides the communication mechanisms of the monitoring service, the EMS service kernel with the implementation of EMS instructions, and the middleware platform that provides the execution of software agents.

The communication service middleware is upheld by standard protocols and technologies included in the *Evolution OS*. In the SOA based EMSP implementation (i.e., the Web Service interface), the cSOAP library was used for development, which is appropriate for these devices (cSOAP, 2009). However, some changes have been made to the original cSOAP library due to device limitations (restriction of memory use, proprietary libraries, etc.). These limitations have forced us to replace cSOAP XML parser, LibXML2 (over 1 MB in size), by another adapted XML parser with limited but sufficient functionalities to achieve our objective. Due to cSOAP limitations, only *RPC style* which uses the same protocol analyser used in the Client-Server version has been developed.

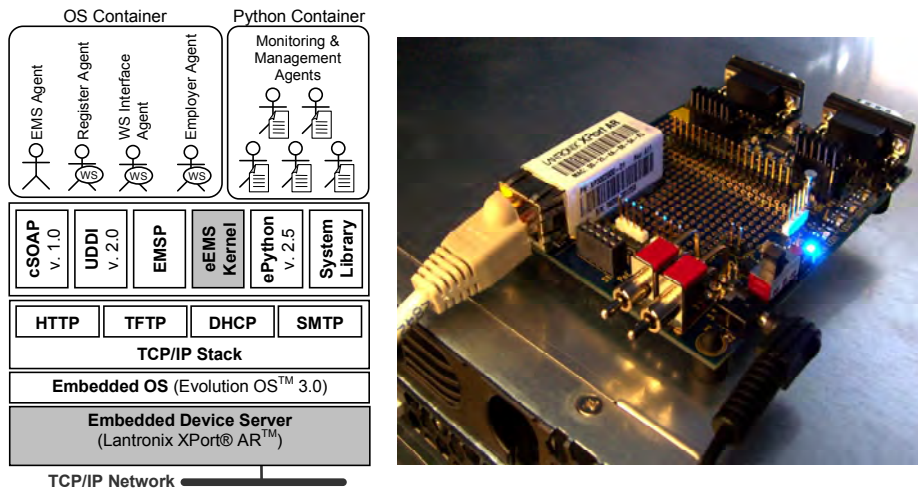


Fig. 4. eEMS device prototype architecture (left) and picture (right).

In addition, in order to register and to publish the services, an UDDI embedded version has been implemented based on UDDI version 2.0 which simply permits publishing the WSDL document associated with the monitoring service.

The EMS service kernel has been implemented as a functions library written in C language and offered as API for the others eEMS device modules. By means of this library, the intrinsic functionalities of the monitoring service are achieved.

In order to implement service agents, a division has been made in the implementation process between static and mobile agents. In the first case, an ad-hoc implementation for the XPort AR device has been developed in C language, using an operative system such as the agents' container. In the second case, in order to establish an execution framework for the mobile agents (the *monitoring agents*), a Python embedded engine (*ePython* version 2.5) has been adapted to the XPort AR features. These *monitoring agents* are implemented as *Python* text scripts.

In order to validate the proposal described in this research work the system and ICT infrastructures that support the Web applications of the Polytechnic University College at the University of Alicante have been chosen like test scenario (Fig. 5). It is a replicated scenario that includes features of high availability and fault tolerance.



Fig. 5. Polytechnic University College Web Site.

The Web applications provide different services for the students around 9044, for the professors around 609, for the administration and services staff and for the external users. These applications are available during 24 hours per day and 7 days per week (inscription system, Web storage system, Web email system, management system, virtual classroom system, general information and others Web applications).

In the table 2, the system components are enumerated, describing the main services included and its infrastructures. This scenario is composed by 10 machines that gives to the users all that them need.

Service Type	Server Model	Number
Apache Web Server	Asus RS120-E4/PA2	3
Apache Tomcat Application Server	Asus RS120-E4/PA2	3
MySQL Database	Asus RS120-E4/PA2	2
OpenLDAP service directory	Asus RS120-E4/PA2	2

Table 2. The Polytechnic University College at the University of Alicante test scenario components.

In figure 6 is showed the chart that include the accesses average of all users to the applications of the Polytechnic University College and the amount of Web traffic transferred in one day. Based on the information displayed in the chart the consumption optimization strategy of the resources has been defined.

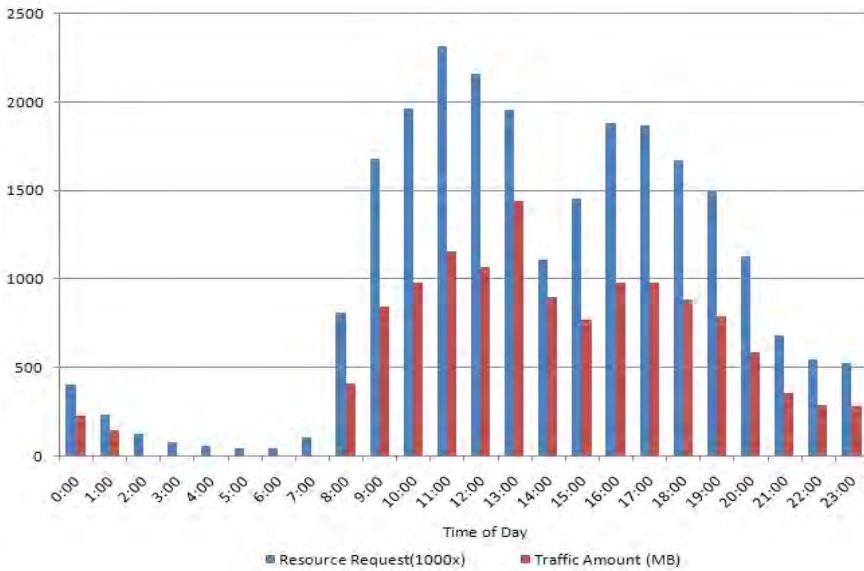


Fig. 6. Average of user requests and amount traffic per day.

In the eEMS device a scheduling has been established that define the time intervals in which all servers have to be power on, also we have considered the traffic by these, due to this variable offers what users needs, and therefore is possible to know when there is more or not information processing into the servers that causes an increment or a diminution of energy consumption. This scheduling has been realized according to the information obtained of the users' accesses to the different applications. In the critical periods the scheduling will obligate to maintain the systems at full performance. Out of the defined periods, the eEMS, in an automatic way, will be responsible of analyzing the information traffic, the request number and accesses to the different applications. In function of the analysis, the eEMS will send the adequate commands sequence in order to power on or power off different system nodes, that is, the system capacity level will be maintained in a dynamic way based on the petition.

The eEMS is able to manage all of the machines that take part into the infrastructure; the number of machines that is power on depends of the traffic that is generated by the users at the time of day. In our scenario there is always 7 machines turn it on due to the system needs to give support to critical applications, however there is several time of day that the eEMS systems keep power off some machines. In a normal infrastructure, there is always 10 machines that are power on and some machines are not been using by the users for that reason the energy consumption is higher. The eEMS allows to use the system in a more efficient way obtaining energy consumption saving. During one week several tests have been realized using the management service and as a result a 13,7% reduction of the energy consumption has been observed in relation to the system without the eEMS device (see table 3 and 4).

Service Type	Server Model	Energy Consumption Average with EMS (wh)		
		Minimum	Average	Maximum
Apache Web Server	Asus RS120-E4/PA2	195,04	660,87	885
Apache Tomcat Application Server	Asus RS120-E4/PA2	195,04	603,79	885
MySQL Database	Asus RS120-E4/PA2	195,04	466,67	590
OpenLDAP service directory	Asus RS120-E4/PA2	97,52	359	590

Table 3. Energy Consumption with the EMS system.

Service Type	Server Model	Energy Consumption Average without EMS (wh)		
		Minimum	Average	Maximum
Apache Web Server	Asus RS120-E4/PA2	292,56	700	885
Apache Tomcat Application Server	Asus RS120-E4/PA2	292,56	700	885
MySQL Database	Asus RS120-E4/PA2	195,04	466,67	590
OpenLDAP service directory	Asus RS120-E4/PA2	195,04	466,67	590

Table 4. Energy Consumption without EMS system.

The energetic saving has not been better (see figure 7) because in this scenario there was one requirement of faults tolerance that obligate to have, minim, two servers to support each service. Obviously, if the system is more complex and there are more replicated nodes for each service the energetic saving will be greater.

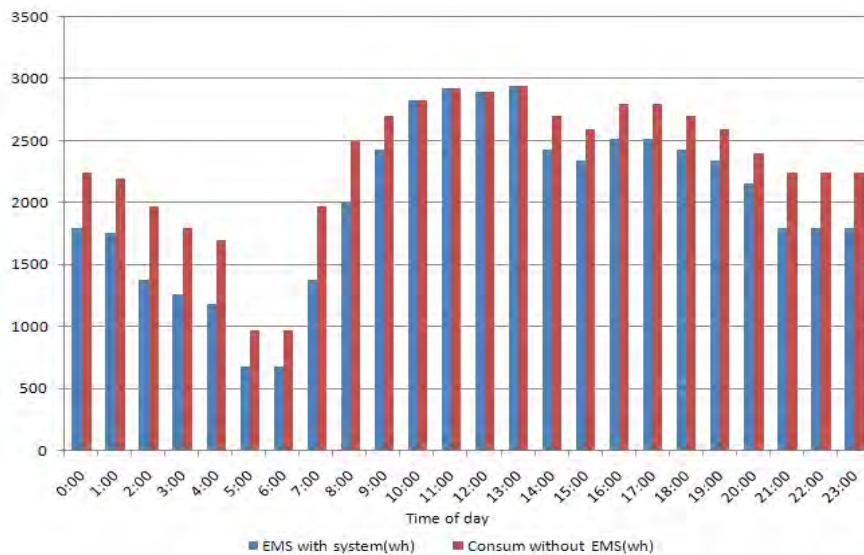


Fig. 7. Relation between energy consumption with the EMS system and without it.

Also, we considerer to highlighted, that the embedded device chosen include the PoE technology, when the eEMS is included in the system its consumption is practically negligible. If the network infrastructures where the eEMS is connected do not support PoE technology, the consumption of *XPort AR* where the service EMS is included would be only 0,957W.

7. Conclusion

In this paper we have presented an energy management system for the ICT infrastructures designed to saving the energy consumption. This system is totally complementary with others approaches oriented to the energy saving and is enough flexible to adapt to different scenarios. One of the most relevant aspects of this system consists of providing these embedded management services in network devices with small size, simple, low power consumption, adjusted costs, autonomous, designed with safety criteria and robustness, and compatible with the traditional network services through the standard protocols such as: SOAP, SMTP or HTTP. In order to validate the proposal, a functional prototype has been designed and implemented. The prototype has been used in a real scenario where we have obtained satisfied results.

We are currently working with other embedded network services and integrating them all in a model based on Semantic Web Services, so that in future they will not only be compatible with existing services, but also with new services or setups which were not considered in the initial design.

8. Acknowledgments

This work was supported by the Spanish Ministry of Education and Science with Grant TIN2006-04081.

9. References

- Beini, L; Boglio, A.; Cavalluci, S. & Ricc . B. (1998). Monitoring system activity for OS-directed dynamic power management. *International Symposium on Low Power Electronics and Design*. ISLPED'98 pp: 185 – 190, 1998 ISBN: 1-58113-059-7.
- Commission European Report: Addressing the challenge of energy efficiency through Information and Communication Technologies, COM (2008) 241 final, Available from <http://ec.europa.eu>
- cSOAP: <http://csoap.sourceforge.net/> (URL).
- Deuty, S. (2004). Exploring the options for distributed and point of load power in telecomm and network applications. *Telecommunications Energy Conference, 2004*. INTELEC 2004. 26th Annual International, pp 223- 229, ISBN: 0-7803-8458-X Chicago, September 2004, United States of America.
- Du, T.C.; Li, E.Y. & Chang, A.P. (2003). Mobile Agents in Distributed Network Management. In *Communications at the ACM*, 46(7), pp127-132. ISSN:0001-0782, New York, July 2003, United Sates Of America.
- Energy Star: <http://www.energystar.gov/> (URL)

- European Union. (2008). Addressing the challenge of energy efficiency through Information and Communication Technologies. <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2008:0241:FIN:EN:PDF> (URL)
- Gartner press release: Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions. Gartner Symposium/ITxpo 2007 Emerging Trends, April 26, (2007) Available from <http://www.gartner.com/it/page.jsp?id=503867>
- Guo, J.; Liao, Y. & Parviz, B. (2005). An Agent-Based Network management system. *Presented at the 2005 Internet and Multimedia Applications.*
- Jammes, F.; Smit, H.; Martinez-Lastra, J.L. & Delamer, I.M. (2005). Orchestration of Service-Oriented Manufacturing Processes. *Proc. of the 10th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2005*, ISBN 0-7803-9401-1, Catania, September 19-22, 2005, Italy
- Lawton, G. (2007). Powering Down the Computing Infrastructure. *Computer*, vol. 40, no. 2, pp. 16-19, IEEE Computer Society, ISSN: 0018-9162.
- Lien, C.H.; Bai, Y.W.; Lin, M.B. & P.-A. Chen. (2004) The saving of energy in web server clusters by utilizing dynamic sever management. *Proceedings. 12th IEEE International Conference on Networs.* vol. 1, pp. 253-257. ISBN: 0-7803-8783-X. Hyderabad, December 2004, India
- Lien, C.H.; Bai, Y.W. & Lin, M. B. (2007). Estimation by Software for the Power Consumption of Streaming Media Servers. *IEEE Transactions on Instrumentation and Measurement.* vol.56 no.5, pp: 1859-1870 . ISSN: 0018-9456. Braunschweig, October 2007, Germany
- Mines, C.; Ferrusi, C.; Brown, E.; Lee, C. & Van-Metre, E. (2008).: The dawn of green IT services. A market overview of sustainability consulting for IT organizations. Forrester Research Report. (2008)
- MON: <http://www.kernel.org/software/mon/> (URL)
- MONIT: <http://www.tildeslash.com/monit/> (URL)
- MUNIN: <http://munin.projects.linpro.no/> (URL)
- NAGIOS: <http://nagios.org> (URL)
- Moshnyaga, G. V. & Tamaru, K. (1997). Energy Saving Techniques for Architecture Design of Portable Embedded Devices. *10Th annual IEEE International ASIC Conference and Exhibit.* ISBN: 0-7803-4283-6, New York, September 1997, United States of America.
- nPULSE: http://www.horsburgh.com/h_npulse.html (URL)
- Pietilainen, J. (2003). Improved Building Energy Consumption with the Help of Modern ICT. *ICEBO. International Conference for Enhanced Building operations.* California, October 2003, United States of America.
- Ren, Z.; Krogh, B. H. & Marculescu, R. (2005). Hierarchical adaptive dynamic power management. *IEEE Transactions on Computers*, vol. 54, no. 4, pp. 409-420. ISSN:0018-9340.
- RFC Project: <http://www.rfc.net> (URL)
- The Green Grid: <http://www.thegreengrid.org/> (URL)
- Topp, U.; Muller, P.; Konnertz, J. & Pick, A. (2002). Web based Service for Embedded Devices, *LNCSS* vol. 2593, 2002, pp. 141-153. ISBN 978-3-540-00745-6